International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 7 Issue 2, March-April 2023 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Attribute Based Ecdsa Searchable Encryption in Cloud Computing (ECDSA - Elliptic Curve Digital Signature Algorithm)

Eben Paul Richard[1], Aravind Swaminathan[2]

[1]ME Student, Department of Computer Science and Engineering,
[2]Professor, Department of Computer Science and Engineering,
[1, 2]Francis Xavier Engineering College, Tirunelveli, Tamil Nadu, India

ABSTRACT

To obtain information from two sources, we need minutiae positions from one source, the orientation from one source, and the reference points from both sources. A combined minutiae template is created with the use of obtained information. In the field of remote data management services, cloud storage has emerged as a major sector. It raises security issues because encryption is currently the greatest method for avoiding data leakage. A promising method among these is public key encryption with keyword search (PKSE), which allows users to quickly search through encrypted data files. The client first generates a search token when to query data files, The cloud server uses the search token to proceed the query over encrypted data files. However, a serious attack is raised when PKSE meets cloud. The problem is Cloud Server discover the privacy information. In our proposed system propose a forward secure Attribute based Elliptic Curve Digital Signature searchable encryption scheme. Finally, our experiments show our scheme is efficient.

KEYWORDS: *Encryption key, token, public key encryption with keyword search (PKSE), Elliptic Curve Digital Signature searchable encryption (ECDSA)*

How to cite this paper: Eben Paul Richard | Aravind Swaminathan "Attribute Based Ecdsa Searchable Encryption in Cloud Computing (ECDSA - Elliptic Curve Digital Signature Algorithm)" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-7 | Issue-2, April 2023, pp.551-556, URL: www.ijtsrd.com/papers/ijtsrd55045.pdf

IJTSRD55045

I. INTRODUCTION:

Cryptographic techniques have been seen as a long-established approach to alleviate the concerns which advocate that data files should be encrypted before outsourcing. Searchable encryption is a cryptographic primitive that allows to execute search operations over encrypted data files. The former is known as symmetric searchable encryption, although it enjoys high efficiency in search process. It provides a terrible performance in data sharing for its complicated secret key distribution. since clients need to share the secret key which will be used for decryption when sharing an encrypted data file to others.

In public key searchable encryption, a client's public key can be used by others to encrypt a data file shared to the client, and client can use its secret key to generate search tokens for its queries. The server can use a search token to test whether an encrypted data file matches the query corresponding to the search token while learning nothing about the query.

In our proposed solution propose a new public key searchable encryption scheme which can achieve forward security A promising method among these is public key encryption with keyword search (PKSE), which allows users to quickly search through encrypted data files. The client first generates a search token when to query data files, The cloud server uses the search token to proceed the query over encrypted data files. However, a serious attack is raised when PKSE meets cloud. The problem is Cloud Server discover the privacy information. In our proposed system propose a forward secure Attribute based Elliptic Curve Digital Signature searchable encryption scheme. Finally, our experiments show our scheme is efficient.

II. EXISTING METHODOLOGY:

In our existing system implement two fold approach (AES-BRS) for data security in Edge computing.

Besides, the processing of data is followed with AES-BRS (Advanced Encryption Standard-Binary Reed-Solomon) code, which is a kind of coding methods based on Reed-Solomon Code. Based on this method, data to be stored is divided into k parts, each of which is l in size, and we generate an encoding part by encoding a matrix, where n = k + m, in which n is the total number of data blocks and m 185 is the number of redundant block. Each encoding part is stored in a storage node. When the number of encoding parts is less than m, the system can repair all the data from any of the k encoding parts. Obviously, no one can recover data as long as they use fewer than k blocks of encoding. Besides physical isolation, we have added several layers of protection to our data to ensure data privacy.

III. PROPOSED METHODOLOGY

In our proposed system propose a forward secure Attribute based Elliptic Curve Digital Signature searchable encryption scheme, in which a cloud server cannot learn any information about a newly added encrypted data file containing the keyword that previously queried. The system model consists of three entities: Clients, Cloud Server and System Clock. With cloud storage, a client may prefer to store its data files into a cloud server for releasing from a large number of data management tasks or sharing its data files to others by using a cloud server. In order to preserve the privacy, a data file should be encrypted before uploading. To search data files from cloud server, a client generates a search token for the querying keyword and sends the search token to cloud server. Upon receiving a search token, the cloud server can search the encrypted data files to return results. It reduce the time complexity. It can greatly reduce the privacy information leaked to a cloud server.

IV. ARCHITECTURAL DESIGN

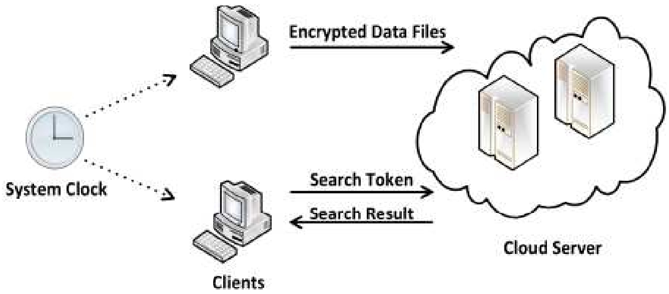

Fig1: System Architecture Diagram

V. SYSTEM ANALYSIS AND DESIGN

In the security model, clients are assumed to be honest, which will honestly perform the protocols. The system clock is a fully trusted entity which will always honestly tell clients the current time. The cloud server is assumed to be honest but curious, which will honestly store encrypted data files and execute the proposed protocols, but curious about the content of data files and queries, namely the cloud server attempts to infer the private information of queries and data files. With the assumptions, the security means the cloud server could learn nothing beyond the test results in search phase n the security model, clients are assumed to be honest, which will honestly perform the protocols. The system clock is a fully trusted entity which will always honestly tell clients the current time. The cloud server is assumed to be honest but curious, which will honestly store encrypted data files and execute the proposed protocols, but curious about the content of data files and queries, namely the cloud server attempts to infer the private information of queries and data files. With the assumptions, the security means the cloud server could learn nothing beyond the test results in search phase. The proposed Attribute based elliptic curve digital signature searchable encryption in cloud storage is depicted in Figure 1. The model is divided into four module such as Client, Cloud Server, Attribute based Elliptic Curve Digital Signature searchable encryption, and clock.

A. Clients:

The entity has large data files to be stored in the cloud server, and also has the requirement of retrieving data files from the cloud server. To search data files from cloud server, a client generates a search token for the querying keyword and sends the search token to cloud server. Upon receiving a search token, the cloud server can search the encrypted data files to return results.

B. Cloud Server:

The entity owns rich storage and computation resources, provides cloud storage services to its clients.

C. Attribute based Elliptic Curve Digital Signature searchable encryption:

The Elliptic Curve Digital Signature Algorithm (ECDSA) is a Digital Signature Algorithm (DSA) which uses keys derived from elliptic curve cryptography (ECC). It is a particularly efficient equation based on public key cryptography (PKC). ECDSA is used across many security systems, is popular for use in secure messaging apps, and it is the basis of Bitcoin security (with Bitcoin "addresses" serving as public keys). ECDSA is also used for Transport Layer Security (TLS), the successor to Secure Sockets Layer (SSL), by encrypting connections between web browsers and a web application. The encrypted connection of an HTTPS website, illustrated by an image of a physical padlock shown in the browser, is made through signed certificates using ECDSA. A main feature of ECDSA versus another popular algorithm, RSA, is that ECDSA provides a higher degree of security with shorter key lengths. This increases its ROI further as ECDSA uses less computer power than RSAm a less secure competing equation.

D. Elliptic Curves

Many readers may associate the term “elliptic” with conic sections from distant school days. An ellipsis is a special case of the general second-degree equation ax2 + bxy + cy2 + dx + ey + f = 0.22 Depending on the values of the parameters a to f, the resulting graph could be a circle, hyperbola, or parabola. Elliptic curve cryptography uses third-degree equations. The DSS defines two kinds of elliptic curves for use with ECC: pseudo-random curves, whose coefficients are generated from the output of a seeded cryptographic hash function; and special curves, whose coefficients and underlying field have been selected to optimize the efficiency of the elliptic curve operations. Pseudo-random curves can be defined over prime fields GF(p) as well as binary fields GF(2m).A prime field is the field GF(p), which contains a prime number p of elements. The elements of this field are the integers modulo p; the field arithmetic is implemented in terms of the arithmetic of integers modulo p.

VI. Mathematical Background

Elliptic curve cryptography involves scalars and points. Typically, scalars are represented with lower-case letters, while points are represented as upper-case letters. Three numerical operations are defined for scalars: addition (+), multiplication (*) and inversion(-1). There are two numerical operations for points: addition (+) and multiplication (×). Although the symbol “+” is used for scalars and points, a point addition follows different rules than the scalar addition. These operations apply to curves over prime fields, as well as curves over binary fields. Algebraic formulae to perform these computations. Computations needed for ECDSA authentication are the generation of a key pair (private key, public key), the computation of a signature, and the verification of a signature. The corresponding equations are found in public literature. Unfortunately, different authors use their own conventions, which makes it difficult to follow their explanations.

A. Key Pair Generation

Before an ECDSA authenticator can function, it needs to know its private key. The public key is derived from the private key and the domain parameters. The key pair must reside in the 23 authenticator’s memory. As the name implies, the private key is not accessible from the outside world. The public key, in contrast, must be openly read accessible.

A few concepts related to ECDSA:

- **Private Key:** A secret number, known only to the person that generated it. A private key is essentially a randomly generated number. In Bitcoin, someone with the private key that corresponds to funds on the blockchain can spend the funds. In Bitcoin, a private key is a single unsigned 256 bit integer (32 bytes).

- **Public Key:** A number that corresponds to a private key, but does not need to be kept secret. A public key can be calculated from a private key, but not vice versa. A public key can be used to determine if a signature is genuine (in other words, produced with the proper key) without requiring the private key to be divulged. In Bitcoin, public keys are either compressed or uncompressed. Compressed public keys are 33 bytes, consisting of a prefix either 0x02 or 0x03, and a 256-bit integer called x. The older uncompressed keys are 65 bytes, consisting of constant prefix (0x04), followed by two 256-bit integers called x and y (2 * 32 bytes). The prefix of a compressed key allows for the y value to be derived from the x value.

- **Signature:** A number that proves that a signing operation took place. A signature is mathematically generated from a hash of something to be signed, plus a private key. The signature itself is two numbers known as r and s. With the public key, a mathematical algorithm can be used on the signature to determine that it was originally produced from the hash and the private key, without needing to know the private key. Resulting signatures are either 73, 72, or 71 bytes long (with approximate probabilities of 25%, 50%, and 25%, respectively-- although sizes even smaller than that are possible with exponentially decreasing probability). ECDSA adopts various concepts in its operation. This involves private keys, public keys and signature. The three features aid in the general operation of the ECDSA. The private key is randomly generated and it is only known to the generating person. Additionally, the key 24 represents a secret number of which the bearer can access funds on a private ledger that correspond to the funds. Contrary to that, the private key can be deployed in the creation of digital signatures on varied data that take in use the digital data algorithm. However, in Bitcoin, the private key is 32 bytes which a composition of 256 unsigned bit integer. On the other hand, a public key is a number that is usually in correspondence to the private key. However, it does not necessarily need to be kept a secret. Additionally, a calculation can be carried out from the private key to determine a public, but the inverse is not possible. A public key is mainly used in the determination of the genuineness of a signature (Snifikino, 2014). However, this process does not necessitate for the divulging of the private key. Bitcoin provides two types of public keys which can either be compressed or uncompressed keys. The signature refers to a number that acts as proof of a signing operation. The generation of the signature is done mathematically from a private key and a hash of what is to be signed. A mathematical algorithm along with the public can be implemented on the signature in the determination of its originality, that is, its generation from a private key and a harsh. A digital signature provides an opportunity for vouching for any messages.

The main benefit of Elliptic Curve Digital Signature Algorithm is that the party authenticating the peripheral is relieved from the constraint to securely store a secret. The authenticating party can authenticate thanks to a public key that can be freely distributed. Authentication ICs, such as those among Maxim's Deep Cover embedded security solutions; help simplify implementation of robust challenge-response authentication methods that form the foundation of more effective application security. The ECDSA authenticators also enable easier authentication of goods from third parties or subcontractors.

VII. ALGORITHM

Elliptic curve algorithms work in a cyclic subgroup of an elliptic curve over a finite field. Therefore, the algorithms will need the following parameters:

The prime pp that specifies the size of the finite field.

The coefficients aa and bb of the elliptic curve equation.

The base point GG that generates our subgroup.

The order nn of the subgroup.

The cofactor hh of the subgroup.

In conclusion, the domain parameters for our algorithms are the sextuple

(p,a,b,G,n,h)(p,a,b,G,n,h).

The algorithm performed by Alice to sign the message works as follows:

Take a random integer kk chosen from {1,...,n−1}{1,...,n−1} (where nn is still the subgroup order). Calculate the point P=kGP=kG (where GG is the base point of the subgroup).

Calculate the number r=xPmodnr=xPmodn (where xPxP is the xx coordinate of PP).

If r=0r=0, then choose another kk and try again.

Calculate

s=k−1 (z+rdA) modns =k−1 (z+rdA) modn (where dAdA is Alice's private key and k−1k−1 is the multiplicative inverse of kk modulo nn).

If s=0s=0, then choose another kk and try again. The pair (r,s)(r,s) is the signature. In plain words, this algorithm first generates a secret (kk). This secret is hidden in rr thanks to point multiplication (that, as we know, is "easy" one way, and "hard" the other way round). rr is then bound to the message hash by the equation s=k−1(z+rdA)modns=k−1(z+rdA)modn.

Note that in order to calculate ss, we have computed the inverse of kk modulo nn. This is guaranteed to work only if nn is a prime number. If a subgroup has a non-prime order, ECDSA can't be used.It's not by chance that almost all standardized curves have a prime order, and those that have a non-prime order are unsuitable for ECDSA. The dynamically add data files to the cloud server, after receiving a new added encrypted data file, the cloud server can immediately know whether the data file matches a previous query by using the search tokens it has received, which can lead to privacy leakage of the new added encrypted data file. Moreover, since keyword space is actually much smaller than password space, if the cloud server has received enough search tokens, it may easily classify a new added 27 encrypted data file by using the search tokens it has received to test the encrypted data file, and then can infer the search token that matches the most data files is corresponding to the frequently used keyword. Therefore, in Attribute based Elliptic Curve Digital Signature searchable encryption schemes, we need forward security, which means a search token cannot be used to search the encrypted data files that produced after the time period of generating the search token (e.g., a search token generated at a time period t cannot be used to search a encrypted data file generated at a time period t1. In general, considering the practicality, most of cloud storage systems can support the function of dynamically adding data files. However, in searchable encryption mechanism, the simple operation of adding data files can seriously lead to the leakage of some privacy information. This is, as clients can dynamically add data files to the cloud server, after receiving a new added encrypted data file, the cloud server can immediately know whether the data file matches a previous query by using the search tokens it has received, which can lead to privacy leakage of the new added encrypted data file. Moreover, since keyword space is actually much smaller than password space, if the cloud server has received enough search tokens, it may easily classify a new added encrypted data file by using the search tokens it has received to test the encrypted data file, and then can infer the search token that matches the most data files is corresponding to the frequently used keyword.

Therefore, in public key searchable encryption schemes, we need forward security, which means a search token cannot be used to search the encrypted data files that produced after the time period of generating the search token. In the security model, clients are assumed to be honest, which will honestly perform the protocols. The system clock is a fully trusted entity which will always honestly tell clients the current time. The cloud server is assumed to be honest but curious, which will honestly store encrypted data files and execute the proposed protocols, but curious about the content of data files and queries, namely the cloud server attempts to infer the private information of queries and data files. With the assumptions, the security means the cloud server could learn nothing beyond the test results in search phase. From the search algorithm, it is easy to know that a search token cannot be used to test an encrypted data file generated after the search token (including cannot be used to test an encrypted data generated at the same time with the search token).

VIII. RESULT AND DISCUSSION

This section presents the results of proposed system which is implemented in Java and MYSQL backend. The project is implemented in Netbeans8.2 and Intel Pentium IV 2.80 GHz Operating system is utilised Moreover, the own dataset is taken for analyzing the performance of the techniques. Java is a programming language and a platform. Java is a high level, robust, object-oriented and secure programming language.

IX. CONCLUSION

The forward security for public key searchable encryption, which means a new added encrypted data file cannot be searched by the search tokens generated before the encrypted data file. This security is urgently required for the public key searchable encryption schemes deployed in cloud storage, and can greatly reduce the privacy information leaked to a cloud server. Finally, our proposed scheme in terms of encryption, token generation and search.

X. FUTURE WORK

We can see the future of Cloud computing as a combination of cloud-based software products and on-premises compute which will help to create hybrid IT solutions. The modified cloud is scalable and flexible, which will provide security and control over data center. With the help of the Internet of Things, the quality of the internet can be increased. With the help of the IoT and Cloud Computing, we can store data in the cloud, for further analyze & provide enhanced performance. The users expect high-quality fast-loading services and application. The network provided will be faster and the ability to receive and deliver that data will be quick.

REFERENCE

[1] Q. Wang, M. Du, X. Chen, Y. Chen, P. Zhou, Chen, and X. Huang, “Privacy-preserving collaborative model learning: The case of word vector training,” IEEE Trans. Knowl. Data Eng., vol. 30, no. 12, pp. 2381–2393, Dec. 2018

[2] H. Zhong, W. Zhu, Y. Xu, and J. Cui, “Multi-authority attribute based encryption access control scheme with policy hidden for cloud storage,” Soft Comput., vol. 22, no. 1, pp.243–251, 2018.

[3] S. Sun, X. Yuan, J. K. Liu, R. Steinfeld, A. Sakzad, V. Vo, and S. Nepal, “Practical backward-secure searchable encryption from symmetric puncturable encryption,” in Proc. ACM Conf. Comput. Commun. Security, 2018, pp. 763–780

[4] P. Xu, S. He, W. Wang, W. Susilo, and H. Jin, “Lightweight searchable public-key encryption for cloud-assisted wireless sensor networks,” IEEE Trans. Ind. Informat., vol.14, no. 8, pp. 3712–3723, Aug. 2018.

[5] H. Yin, J. Zhang, Y. Xiong, L. Ou, F. Li, S. Liao, and K. Li, “CPABSE: A ciphertext-policy attribute-based searchable encryption scheme,” IEEE Access, vol. 7, pp. 5682–5694, 2019.

[6] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, “Practical attributebased multi-keyword search scheme in mobile crowdsourcing,” IEEE Internet Things J., vol. 5, no. 4, pp.3008–3018, Aug. 2018.

[7] Ning, J., Huang, X., Susilo, W., Liang, K., Liu, X., & Zhang, Y. (2020). Dual Access Control for Cloud-Based Data Storage and Sharing. IEEE Transactions on Dependable and Secure Computing, 1–1. doi:10.1109/tdsc.2020.3011525

[8] P. Chinnasamy and P. Deepalakshmi, "Design of Secure Storage for Health-care Cloud using Hybrid Cryptography," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 1717-1720, doi:10.1109/ICICCT.2018.8473107.49

[9] Prabhu kavin, B., & Ganapathy, S. (2019). A secured storage and privacy-preserving model using CRT for providing security on cloud and IoT-based applications. Computer Networks, 151, 181–190. doi:10.1016/j.comnet.2019.01.032

[10] Bhardwaj, F. Al-Turjman, M. Kumar, T. Stephan and L. Mostarda, "Capturing-the-Invisible (CTI): Behavior-Based Attacks Recognition in IoT-Oriented Industrial Control Systems," in IEEE Access, vol. 8, pp. 104956-104966, 2020, doi:10.1109/ACCESS.2020.2998983.